**JAVASCRIPT WEEK 8 SUMMARY**

**Day 1 – Introduction to Node JS**

**What is Node JS? -** Node.js is an open-source, server-side runtime environment that allows developers to run JavaScript code on the server. It is built on the V8 JavaScript engine from Google Chrome, which compiles JavaScript into machine code for faster execution. Node.js provides a non-blocking, event-driven architecture, making it well-suited for building scalable and high-performance network applications.

**How is Node JS initiated on a computer? –** Node.js is initiated on a computer by installing it and running JavaScript files using the Node.js runtime.

**Why do we use Node JS? -** Node.js is a versatile and powerful runtime environment that is used for building a wide range of applications, from web servers and APIs to real-time applications and microservices. Its efficiency, scalability, and extensive ecosystem make it a popular choice among developers for various development projects.

**What can Node JS do?**

* Server-side development – Node.js are commonly used to build web servers, APIs and backend services for web applications. It can handle HTTP requests and serve web content efficiently.
* Real-time applications - Node.js excels in building real-time applications, such as chat applications, online gaming servers, and collaborative tools, due to its non-blocking, event-driven architecture.
* Microservices - It's suitable for developing microservices-based architectures, allowing you to create small, independent services that communicate via APIs.
* Cross-Platform - Node.js is compatible with various operating systems, making it easy to deploy applications across different environments, including Windows, macOS, etc.
* Single Programming Language - Developers can use JavaScript for both frontend and backend development, promoting code reuse and consistency across the entire stack.
* Efficient I/O Handling - Node.js efficiently manages asynchronous I/O operations, making it ideal for tasks involving concurrent connections and high-performance computing.
* Large Ecosystem - Node.js boasts a vast ecosystem of open-source packages and libraries available via npm, simplifying the integration of third-party modules into projects.
* Scalability - Node.js applications can be horizontally scaled by adding more server instances to handle increased traffic, ensuring scalability for large applications.
* Real-time communication - Node.js supports technologies like WebSockets, enabling real-time, bidirectional communication between clients and servers.
* Community and support - Node.js benefits from a vibrant and active developer community, offering ample resources, tutorials, and support for developers.
* Performance - Node.js leverages the high-performance V8 JavaScript engine from Google Chrome, compiling JavaScript code into machine code for speedy execution.
* Cost-effective - Using Node.js can be cost-effective for businesses as it allows developers to work with a single language for both frontend and backend development, reducing the need for expertise in multiple languages.

**Is a module in Node JS the same as in JavaScript? –** While Node.js modules and JavaScript modules share similar goals of organizing and reusing code, they have different syntax and mechanisms for defining and importing modules.

**What is NPM? -** NPM stands for "Node Package Manager." It is a widely-used package manager for Node.js, which is a runtime environment for executing JavaScript code on the server-side. NPM serves as a central repository for publishing, discovering, and managing reusable JavaScript libraries and packages, making it an essential tool in the Node.js ecosystem.

**What is contained in a Node JS Package? –** Here’s what you can find in a Node.js package:

* JavaScript Code
* Package configuration
* Dependencies
* Tests
* Documentation
* Examples
* Configuration Files
* Assets
* License Information
* Contributor Guidelines
* Build and Distribution Files
* Git Repository